

# Write-Up Tutorial 2 APAP

Nama : Lintang Matahari Hasani

NPM : 1506689231

Kelas : A

## Project Hello World

Tampilan method index()

```
10 @Controller
11 public class PageController {
12     @RequestMapping("/hello")
13     public String index() {
14         return "hello";
15     }
16 }
```

Tampilan template hello.html

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4 <title>First Page</title>
5 </head>
6 <body>
7 <h1>Hello, World!</h1>
8 </body>
9 </html>
```

## Latihan Project Hello World

- 1) Ganti baris tersebut menjadi `@RequestMapping("/hello123")`

```
10 @Controller
11 public class PageController {
12     @RequestMapping("/hello123")
13     public String index() {
14         return "hello";
15     }
16 }
```

Pertanyaan: *Apakah compile error?*

Tidak terjadi compile error. Akan tetapi, ketika Spring Boot yang sedang berjalan dihentikan dan kembali dijalankan, muncul pesan error pada browser seperti pada gambar di bawah ini:



- 2) Ganti nama method index() dengan nama method hello()

```
10 @Controller
11 public class PageController {
12     @RequestMapping("/hello")
13     public String hello() {
14         return "hello";
15     }
16 }
```

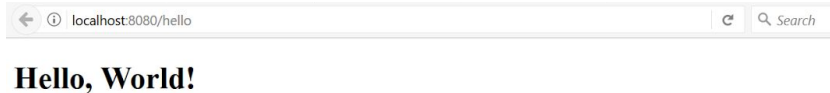
Pertanyaan: *Apakah compile error?*

Tidak terjadi compile error. Ketika Spring Boot dihentikan dan dijalankan kembali pesan "Hello, World!" tetap ditampilkan sebagaimana sebelum dilakukan perubahan nama method.

Nama : Lintang Matahari Hasani

NPM : 1506689231

Kelas : A

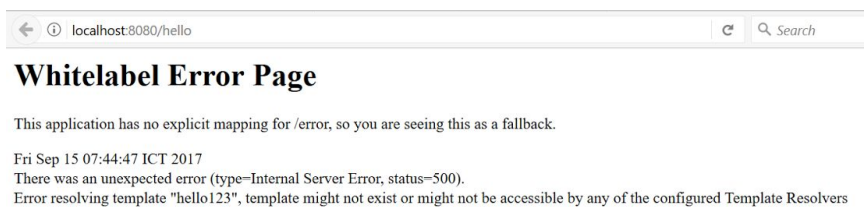


3) Ganti string return type menjadi return "hello123";

```
10 @Controller
11 public class PageController {
12     @RequestMapping("/hello")
13     public String index() {
14         return "hello123";
15     }
16 }
```

Pertanyaan: *Apakah compile error?*

Tidak terjadi compile error. Akan tetapi setelah Spring Boot dihentikan dan dijalankan kembali, muncul pesan error pada browser seperti pada gambar berikut



Pertanyaan: *Menandakan apakah String yang di return tersebut?*

String yang dikembalikan oleh method tersebut adalah nama template yang terdapat pada folder templates. Error pada kasus diatas disebabkan oleh tidak adanya nama template "hello123" pada folder templates.

## Request Parameter (Query String)

Tampilan method greeting

```
17 @RequestMapping("/greeting")
18 public String greeting (@RequestParam(value = "name") String name, Model model) {
19     model.addAttribute("name", name);
20     return "greeting";
21 }
22 }
```

Tampilan template greeting.html

```
1 <!DOCTYPE html>
2 <html xmlns:th = "http://www.thymeleaf.org">
3 <head>
4 <title> Second Page </title>
5 </head>
6 <body>
7 <p th:text = "Selamat datang " + ${name} + "!">Sapaan untuk user</p>
8 </body>
9 </html>
```

## Latihan Request Parameter (Query String)

- 1) Ubah nilai anotasi RequestMapping dari "/greeting" menjadi "/hello/greeting"

```
17 @RequestMapping("/hello/greeting")
18 public String greeting (@RequestParam(value = "name") String name, Model model) {
19     model.addAttribute("name", name);
20     return "greeting";
21 }
```

Buka localhost: 8080/hello/greeting?name=chanek

localhost:8080/hello/greeting?name=chanek

Selamat datang chanek!

Pertanyaan: *apakah hasilnya?*

Pada browser ditampilkan String "Selamat datang chanek!"

- 2) Akses localhost: 8080/greeting

Pertanyaan: *Apakah hasilnya?*

localhost:8080/hello/greeting

### Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Fri Sep 15 08:00:18 ICT 2017

There was an unexpected error (type=Bad Request, status=400).

Required String parameter 'name' is not present

Terjadi error. Hal ini disebabkan oleh kosongnya parameter String 'name' dan method tidak menangani hal tersebut.

Ubah header method menjadi seperti ini:

```
17 @RequestMapping("/greeting")
18 public String greeting (@RequestParam(value = "name", required = false) String name, Model model) {
19     model.addAttribute("name", name);
20     return "greeting";
21 }
```

Stop Spring Boot yang sedang berjalan, run kembali, buka localhost: 8080/greeting

Pertanyaan: *Apakah hasilnya?*

localhost:8080/greeting

Selamat datang null!

Pada browser ditampilkan string “Selamat datang null”. Kekosongan parameter String ‘name’ telah ditangani oleh method dan nilainya menjadi null.

3) Ubah header method menjadi seperti ini:

```
17 @RequestMapping("/greeting")
18 public String greeting (@RequestParam(value = "name", required = false,
19 defaultValue = "dunia") String name, Model model) {
20     model.addAttribute("name", name);
21     return "greeting";
22 }
```

Stop Spring Boot yang sedang berjalan, run kembali, dan buka localhost:8080/greeting.

Pertanyaan: *apakah hasilnya?*



Pada browser ditampilkan String “Selamat datang dunia!”. Pada saat parameter String ‘name’ kosong/bernilai null, parameter tersebut akan diisi dengan default value String “dunia”.

4) Perhatikan bahwa pada berkas greeting.html, tag paragraf yang kita tambahkan adalah sebagai berikut:

```
6 <body>
7 <p th:text = "'Selamat datang ' + ${name} + '!'">Sapaan untuk user</p>
8 </body>
```

Pertanyaan: *Mengapa tulisan “Sapaan untuk user” tidak pernah muncul?*

Hal tersebut terjadi karena template hanya akan menampilkan text yang berada di th:text jika pada tag tersebut terdapat th:text.

## Path Variable

Tampilan method greetingPath()

```
24 @RequestMapping("/{greeting/{name}}")
25 public String greetingPath(@PathVariable String name, Model model){
26     model.addAttribute("name", name);
27     return "greeting";
28 }
```

Tampilan hasil

localhost:8080/greeting/chanek

Selamat datang chanek!

## Latihan Path Variable

1. Akses localhost:8080/greeting/  
Pertanyaan: *Apa hasilnya?*

localhost:8080/greeting

Selamat datang dunia!

Pada browser akan ditampilkan String “Selamat datang dunia!”. Berdasarkan path yang diakses, method yang dipanggil adalah yang memiliki Request Mapping /greeting yakni method greeting()

Ubah method greetingPath menjadi seperti berikut

```
30 @RequestMapping(value = {"{greeting}", "greeting/{name}"})
31 public String greetingPath(@PathVariable Optional <String> name, Model model){
32     if (name.isPresent()) {
33         model.addAttribute("name", name.get());
34     } else {
35         model.addAttribute("name", "dengklek");
36     }
37     return "greeting";
38 }
39
```

Akses localhost:8080/greeting/  
Pertanyaan: *Apa hasilnya?*

localhost:8080/greeting

### Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Fri Sep 15 08:41:34 ICT 2017

There was an unexpected error (type=Internal Server Error, status=500).

Ambiguous handler methods mapped for HTTP path 'http://localhost:8080/greeting': {public java.lang.String com.example.demo.controller.PageController.greetingPath(java.util.Optional,org.springframework.ui.Model), public java.lang.String com.example.demo.controller.PageController.greeting(java.lang.String,org.springframework.ui.Model)}

Terjadi error karena terjadi ambiguity. Hal ini dikarenakan adanya Request Mapping yang sama antara method greeting() dan greetingPath(). Setelah method greeting() di hapus/dijadikan comment, browser akan menampilkan String “Selamat datang dengklek!”.

localhost:8080/greeting

Selamat datang dengklek!

Nama : Lintang Matahari Hasani

NPM : 1506689231

Kelas : A

Akses localhost:8080/greeting/chanek

Pertanyaan: *Apa hasilnya?*

Bila method Mapping yang ambigu belum diubah seperti pada kasus sebelumnya maka browser akan menampilkan pesan error. Jika Ambiguitas Request Mapping sudah diselesaikan maka browser akan menampilkan String “Selamat datang chanek!”.



## Latihan Perkalian

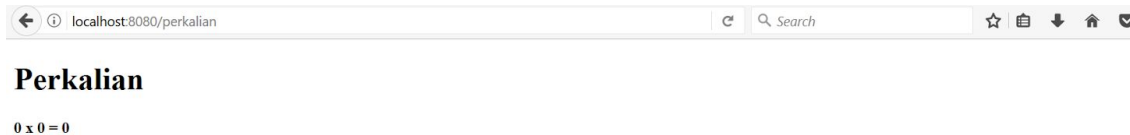
### Tampilan method perkalian()

```
40 @RequestMapping("/perkalian")
41 public String perkalian (
42     @RequestParam(value = "a", required = false, defaultValue = "0") int a, Model model1,
43     @RequestParam(value = "b", required = false, defaultValue = "0") int b, Model model2,
44     @RequestParam(value = "hasil", required = false, defaultValue = "0") int hasil, Model model3) {
45
46     model1.addAttribute("a", a);
47     model2.addAttribute("b", b);
48
49     hasil = a * b;
50
51     model3.addAttribute("hasil", hasil);
52
53     return "perkalian";
54 }
55 }
```

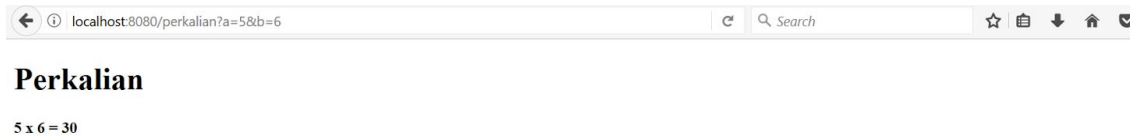
### Tampilan template perkalian.html

```
1 <!DOCTYPE html>
2 <html xmlns:th = "http://www.thymeleaf.org">
3 <head>
4 <title>Perkalian</title>
5 </head>
6 <body>
7 <h1>Perkalian</h1>
8 <h4 th:text = "${a} x ${b} = " + ${hasil}>hasil perkalian</h4>
9 </body>
10 </html>
```

### Kasus 1: default value 0 jika parameter a dan b tidak ada



### Kasus 2: hasil perkalian a dan b jika parameter a dan b ada



### Kasus 3: hasil perkalian 0 jika hanya salah satu parameter a atau b ada



## Lesson Learned

Pada tutorial kali ini saya belajar banyak mengenai framework Spring Boot, khususnya mengenai penggunaan controller untuk menampilkan template. Selain itu, saya juga mempelajari routing pada Spring Boot.

PageController pada Spring Boot merupakan sebuah Controller Class yang berisi method-method yang menerima parameter tertentu jika ada dan mengembalikan nama template untuk ditampilkan pada browser. Singkatnya, Controller merupakan Class yang menerima HTTP Request dan mengembalikan View. View pada Tutorial ini berupa file HTML yang terdapat pada folder template.

Pada setiap method terdapat Request Mapping yang menandakan Request HTTP, misal anotasi `@RequestMapping("/hello")` pada method `hello()` menandakan bahwa method tersebut akan dipanggil setiap ada HTTP Request pada path `/hello`

Untuk method yang menangkap parameter tertentu, anotasi `@RequestParam("nama_parameter")` digunakan. Method kemudian akan menerima parameter tersebut dengan GET request. Parameter kemudian di 'pass' ke view dengan objek Model.